Tema: Amazenagem de dados e configuração de Banco de Dados para desempenho

1. **Introdução**

Ao projetar um banco de dados, você deve certificar-se de que o banco de dados executa todas as funções importantes correta e rapidamente. Alguns problemas de desempenho podem ser resolvidos depois que o banco de dados estiver em produção. Entretanto, outros problemas de desempenho podem ser resultado de um mau projeto de banco de dados, só podendo ser resolvidos pela alteração da estrutura e projeto do banco de dados.

Ao projetar e implementar um banco de dados, você deve identificar as tabelas grandes no banco de dados e os processos mais complexos que o banco de dados executará. Você também deve reservar consideração especial ao desempenho quando projetar essas tabelas. Além disso, você deve considerar o efeito sobre o desempenho ao aumentar o número de usuários que podem acessar o banco de dados.

1. **Amazenagem de dados**
   1. **Formas de Armazenamento**
      1. **Cache** – é a forma de armazenamento mais rápida e dispendiosa. A memória cache é pequena, seu uso é gerenciado pelo HW do sistema de computador.

Memória cache é um tipo de memória de alta velocidade que fica próxima à CPU e consegue acompanhar a velocidade de trabalho da CPU. Por ser uma memória de alta velocidade ela é difícil de ser produzida e por isso mesmo muito cara. Por isso é que raramente encontramos quantidades de memória cache maiores que 1 MB. É bom lembrar que, ao invés da tecnologia DRAM, usa-se a tecnologia SRAM para a produção de chips de memória cache.

Se o processador precisa de uma informação e ela está no cache, ótimo, a informação é acessada e o desempenho é alto. Quando isso acontece, chamamos de cache “hit” ("acerto"). Porém se a informação não estiver no cache, ela vai ter que ser lida da memória RAM “normal”, o que é um processo mais lento. Este caso é chamado de cache “miss” ("erro"). A idéia é fazer com que o número de cache hits seja muito maior que o número de cache misses.

* + 1. **Memória principal** – o meio de armazenamento utilizado para os dados que estão disponíveis para serem operados é a memória principal. Embora a memória principal possa conter muitos megabytes de dados, ou ainda centenas de gigabytes de dados em grandes sistemas de servidor, ela geralmente é muito pequena(ou muito cara) para armazenar o banco de dados inteiro. O conteúdo da memória principal normalmente se perde se houver falha de energia ou falha no sistema. Memória **principal**: também chamadas de memória real, são memórias que o [processador](http://pt.wikipedia.org/wiki/Processador) pode endereçar diretamente, sem as quais o computador não pode funcionar. Estas fornecem geralmente uma *ponte* para as secundárias, mas a sua função principal é a de conter a informação necessária para o processador num determinado momento; esta informação pode ser, por exemplo, os programas em execução. Nesta categoria insere-se a [memória RAM](http://pt.wikipedia.org/wiki/Mem%C3%B3ria_RAM),que é uma memória de semicondutores, volátil, com acesso aleatório, isto é, palavras individuais de memória são acessadas diretamente, utilizando uma lógica de endereçamento implementada em hardware. Também pode-se compreender a [memória ROM](http://pt.wikipedia.org/wiki/Mem%C3%B3ria_ROM) (*não volátil*), [registradores](http://pt.wikipedia.org/wiki/Registrador_%28inform%C3%A1tica%29) e [memórias *cache*](http://pt.wikipedia.org/wiki/Mem%C3%B3ria_cache).
    2. **Memória flash** – A memória flash difere da memória principal porque os dados sobrevivem à falta de energia. A leitura dos dados da memória flash leva menos de 1000 nanossegundos (1 nanossegundo é 1/1000 de um microssegundo), o que é aproximadamente a mesma velocidade da leitura de dados na memória principal. Porém, a escrita dos dados é mais complicada – os dados podem ser escritos uma vez, o que leva cerca de 4 a 10 microssegundos, mas não podem ser escritos diretamente. Para escrever a memória que já foi escrita, temos que apagar o banco de memória ao mesmo tempo, depois, ele estará pronto para ser escrito novamente. Uma desvantagem da memória flash é que ela só pode aceitar um número limitado de ciclos de apagamento, variando de 10.000 a 1 milhão. A memória flash é uma forma de memória somente leitura programável e apagável eletricamente (EEPROM); outras formas de EEPROM permitem que locais de momórias individuais sejam apagadas e reescritos, mas esses tipos não são muito utilizadas. Um modelo de memória flash que permite escrita é a utilizada em câmeras, pendrives USB.
    3. **Armazenamento em disco magnético** – o principal meio para armazenamento de dados on-line a longo prazo é o disco magnético. Normalmente, o banco de dados inteiro é armazenado no disco magnético. O sistema precisa mover os dados do disco para a memória principal, de modo que possam ser acessados. Depois que o sistema tiver realizado as operações designadas, os dados que foram modificados precisam ser gravados em disco.

O tamanho dos discos magnéticos atualmente varia de alguns gigabytes até 4000 gigabytes. O armazenamento em disco sobrevive a faltas de energia e falhas no sistema. Os próprios dispositivos de armazenamento em disco às vezes podem falhar e, portanto, destruir dados, mas essas falhas normalmente ocorre com menos freqüência do que as falhas **do sistema.**

* + 1. **Armazenagem óptica** – as formas mais populares de armazenamento óptico são os CD, DVD, BlueRay. Pode ser utilizado para uso de backup do banco de dados, porém devido a capacidade não é principal meio de armazenamento para esse fim.
    2. **Armazenamento em fita** – O armazenamento em fita é usado principalmente para backup e arquivamento de dados. Embora a fita magnética seja mais barata do que os discos, o acesso aos dados é muito mais lento, pois a fita precisa ser acessada sequencialmente. Ao, contrário, o armazenamento em disco é chamado de armazenamento por acesso direto.
    3. Hierarquia dos dispositivos de armazenamento

Os diversos meios de armazenamento podem ser organizados em uma hierarquia, de acordo com a sua velocidade e seu custo. Os níveis mais altos são os mais caros, mas são mais rápidos. Ao descermos na hierarquia, o custo por bit diminui, enquanto o tempo de acesso aumenta. Essa compensação é razoável; se determinados sistemas de armazenamento fosse mais rápido e mais barato do que outro então não haveria motivo para usar um meio mais caro e mais lento.

Os meios de armazenamento mais rápidos – por exemplo, cachê e memória principal – são considerados armazenamento primário. Os meios no próximo nível da hierarquia são considerados de **armazenamento secundário** ou **armazenamento on-line** – discos ópticos. Já os discos ópticos e as fitas magnéticas são considerados de **armazenamento terciário** ou **armazenamento offline**.

* 1. Medidas de desempenho dos discos

As principais medidas das qualidades de um disco são **capacidade, tempo de acesso, taxa de transferência de dados e confiabilidade**

* + 1. **Tempo de acesso** – é o tempo desde quando uma solicitação de leitura ou escrita é emitida até quando inicia a transferência dos dados.
       1. **Tempo de busca** – em um HD, é o tempo de reposicionar o braço na trilha correta.
       2. **Tempo de latência rotacional** – quando a cabeça atingir a trilha desejada o tempo gasto esperando encontrar o setor a ser acessado.
    2. **Taxa de transferência de dados – é a velocidade em que os dados podem ser apanhados ou armazenados no disco.**
    3. **Confiabilidade** – medida usando o **Tempo média para a falha(MTTF – Mean Time to Failure)** – que é o tempo médio para a falha de um disco. É a quantidade, média, que podemos esperar que o sistema funcione continuamente sem qualquer falha.
  1. Otimização de acesso ao bloco de disco

As solicitações de E/S de um disco são geradas pelo sistema de arquivos e pelo gerenciador de memória virtual, encontrando na maioria dos sistemas operacionais. Um bloco é uma unidade lógica constituída de um tamanho fixo de setores contíguos. Os dados são transferidos entre disco e memória principal em unidades de blocos.

O acesso no disco é de várias ordens de grandezas mais lento do que o acesso aos dados na memória principal. Como resultado, diversas técnicas foram desenvolvidas para melhorar a velocidade de acesso aos blocos do disco.

* + 1. **Escalonamento** – Se vários blocos de um cilindro precisam ser transferidos do disco para a memória principal, podemos economizar tempo de acesso solicitando os blocos na ordem em que passarão sob as cabeças. Se os blocos desejados estiverem em cilindros diferentes, é vantajoso solicitar os blocos em uma ordem que reduza o movimento no braço do disco.

Os algoritmos de escalonamento de braço do disco tentam ordenar os acessos as trilhas em um padrão que aumenta o número de acessos que podem ser processados. O algoritmo normalmente utilizado é o do **elevador.** As controladoras de disco normalmente realizam a tarefa de ordenação dos blocos no disco, da posição rotacional das placas de disco e da posição do braço do disco.

* + 1. **Organização de arquivo** – Para reduzir o tempo de acesso ao bloco, podemos organizar os blocos no disco de uma maneira que corresponda a como esperamos que os dados sejam acessado. Por exemplo, se for acessado sequencialmente, armazenar sequencialmente.

Porém, nem sempre isso é possível, já que pode ocorrer a fragmentação de um arquivo seqüencial. Alguns sistemas possuem ferramentas para organizar os fragmentos no disco, aumentando o desempenho de acesso.

* + 1. **Buffers de escrita não voláteis** – como o conteúdo da memória principal se perde em caso de falta de energia, as informações de sobre atualizações de broadcast precisam ser armazenadas em disco para sobreviver a possíveis falhas. Por esse motivo, o desempenho de aplicações com uso intensivo de atualizações de banco de dados, como sistemas de processamento de transação, depende bastante da velocidade das escritas no disco.

Podemos usar a memória de acesso aleatório não volátil (NVRAM) para agilizar bastante a escrita no disco. Um modo comum de implementar a NVRAM é usar uma RAM com bateria. Pg 299

* + 1. **Disco de log** – outra técnica para reduzir a latência de escrita é usar um disco de log – ou seja, um disco dedicado a escrever um log seqüencial – mais ou menos da mesma maneira que um buffer de RAM não volátil. Todo o acesso ao disco de log é seqüencial, basicamente eliminando o tempo de busca, e vários blocos consecutivamente podem ser escritos ao mesmo tempo, tornando as escritas de log várias vezes + rápidas q as escritas aleatórias. Como antes, os dados também precisam ser escritos em seu local real no disco, mas o disco pode fazer isso mais tarde, sem que o SBD tenha que esperar até que a escrita termine. Além do mais, o disco de log pode reordenar as escritas para reduzir o movimento do braço do disco. Se o sistema falhar antes do final de alguma escrita no local real do disco, quando o sistema retornar, ele lê o disco de log para descobrir as escritas que não foram completadas, executando-as em seguida.

**Recuperaçã o baseada em log**

1. **Configurações para Desempenho**
   1. **Ajustes de Desempenho(parâmetros)**

O ajuste de desempenho de um sistema envolve ajustar vários parâmetros e opções de projeto a fim de melhorar seu desempenho para uma aplicação específica. Diversos aspectos de um projeto de SBD – variando desde aspectos de alto nível, como projeto de esquema até parâmetros de BD, como tamanho de buffer e aspectos de HW.

* + 1. Localização de gargalos – em um sistema bem balanceado, nenhum componente isolado é um gargalo. Se o sistema tiver um gargalo, os componentes que não fazem parte do gargalo são subutilizados e talvez pode ser substituído por outro com menos desempenho.

Os sistemas de BD são modelados como sistemas de enfileiramento, onde cada fila representa uma determinada operação em uma transação, Por exemplo, posso ter uma fila esperando ler o disco, uma fila de processos esperando E/S, fila de bloqueio A utilização de um recurso deverá ser baixa o suficiente para que o tamanho da fila seja curto. As utilizações em torno de 70% são consideradas boas acima de 90% excessiva.

* + 1. **Parâmetros ajustáveis** – As opções para ajustar o sistema nesse nível inclui acrescentar discos ou usar um sistema RAID se a E/S for um gargalo, aumentar a quantidade de memória se o tamanho do buffer de disco for um gargalo ou passar um processador mais rápido se o uso de CPU for um gargalo.

Um segundo nível consiste nos parâmetros do sistema de BD, como tamanho de buffer ou intervalos de pontos de verificação.

O terceiro nível é o mais alto. Inclui o esquema e as transações que são executadas.

* 1. Ajustes de Hardware

Um fator importante é saber se o sistema projetado consegue lidar com a taxa das operações de E/S exigidos pelo BD. Observe que aqui o fator importante é a taxa de transferência, e não a capacidade. O número de operações de E/S pode ser reduzido armazenando os dados em memória.

* + 1. Regra dos 5 min: se uma página é usada com mais freqüência do que uma vez em 5 min, ela deve ser colocada na memória
    2. Regra do 1 min : consideramos que os dados acessados sequencialmente devem ser mantidos em cachê na memória se forem usados pelo menos uma vez em 1 min.
  1. **Ajustando o esquema**
     1. **Ajuste de índices** – se as consultas forem um gargalo, normalmente podemos agilizá-las criando índices apropriados para a relação.

Se as atualizações forem um gargalo, pode haver muitos índices, que precisam ser atualizados quando as relações forem atualizadas. A remoção de índices pode agilizar certas atualizações.

* + 1. **Usando visões materializadas** –

Visão materializada é quando o sistema armazena além da consulta que a define armazena o seu resultado.

a visão materializadas podem agilizar bastante certos tipos de consultas, em particular consultas de agregação

* + 1. Ajuse automatizado do projeto físico – o ajuste precisa ser antes que a aplicação seja totalmente desenvolvida, e o conteúdo real do banco pode ser pequeno no banco de desenvolvimento, mas espera-se que seja muito maior no banco de produção. Assim, algumas ferramentas permitem que seja feito ajuste segundo informações sobre o tamanho esperado do banco e estatísticas relacionadas.

Database Tuning Assistant da Microsoft

* + 1. Ajuste de transações
    - Melhorar a orientação do conjunto
    - Reduzir a disputa pelo bloqueio

Usar arquivos e grupos de arquivos aprimora o desempenho do banco de dados porque permite criar um banco de dados em vários discos, vários controladores de discos ou sistemas RAID. Por exemplo, se o seu computador tiver quatro discos, crie um banco de dados que consista em três arquivos de dados e um arquivo de log, com um arquivo em cada disco. À medida que os dados são acessados, quatro cabeçotes de leitura/gravação podem acessar os dados em paralelo e simultaneamente. Isso acelera as operações de banco de dados.

Adicionalmente, os arquivos e grupos de arquivos ativam o posicionamento de dados porque uma tabela pode ser criada em um grupo de arquivos específico. Isso aprimorará o desempenho porque toda a E/S de uma determinada tabela pode ser dirigida a um disco específico. Por exemplo, uma tabela usada intensamente pode ser posta em um arquivo ou grupo de arquivos localizados em um disco, e as outras tabelas usadas menos intensamente no banco de dados podem ser postas nos outros arquivos do grupo de arquivos localizados em um segundo disco.

A seguir, algumas recomendações gerais para quando se estiver trabalhando com arquivos e grupos de arquivos:

* A maioria dos bancos de dados funcionará bem com um único arquivo de dados e um único arquivo de log de transação.
* Quando vários arquivos forem usados, crie um segundo grupo de arquivos para o arquivo adicional e transforme o grupo de arquivos em padrão do grupo de arquivos. Desse modo, o arquivo primário conterá somente tabelas e objetos do sistema.
* Para maximizar o desempenho, crie arquivos ou grupos de arquivos em tantos discos físicos diferentes e disponíveis quanto possível. Insira objetos que disputem pesadamente por espaço em diferentes grupos de arquivos.
* Use grupos de arquivos para ativar a colocação de objetos em discos físicos específicos.
* Insira tabelas diferentes usadas nas mesmas consultas de junção em diferentes grupos de arquivos. Isto aperfeiçoará o desempenho por conta da busca por dados adicionados pela E/S paralela do disco.
* Insira tabelas excessivamente acessadas e índices não clusterizados que pertençam às tabelas de diferentes grupos de arquivos. Isto aperfeiçoará o desempenho por causa da E/S paralela caso os arquivos estejam localizados em discos físicos diferentes.
* Não insira o arquivo ou arquivos de log de transação no mesmo disco físico que contém os outros arquivos e grupos de arquivos.